Continual Learning

Insights from SplitMNIST

CS 5100 project presentation

Khoury College of Computer Science, Northeastern University

Saurabh Shetty
shetty.sau@northeastern.edu

Siddharth Mittal
mittal.sid@northeastern.edu

TABLE OF CONTENTS

01
The Continual Learning Problem

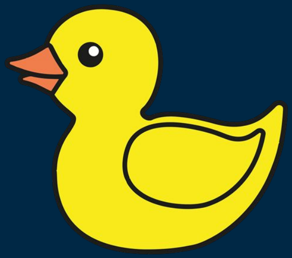
02
Approaches To Continual Learning

03
Experiments & Results

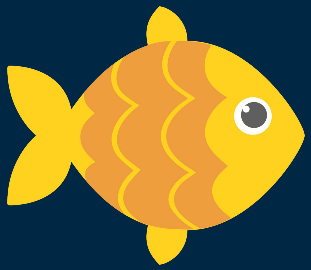
04
Conclusion

The Continual Learning Problem

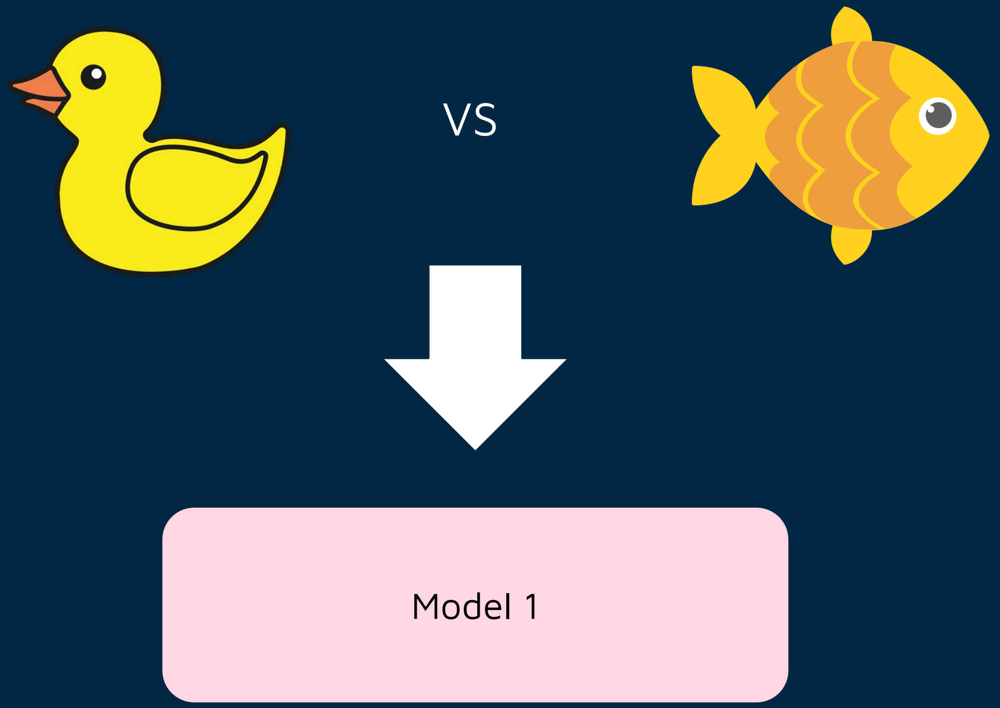
Train a classification model



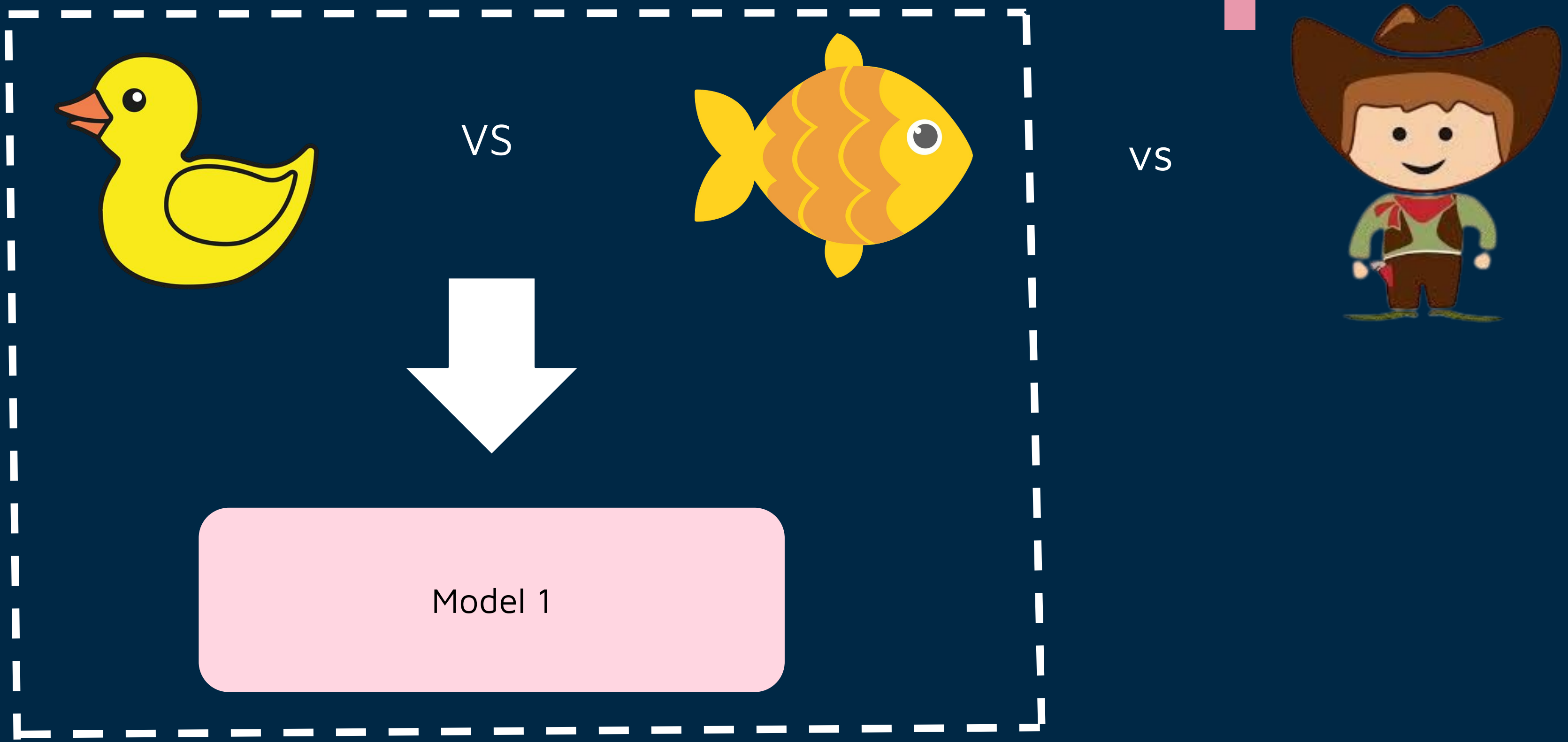
VS



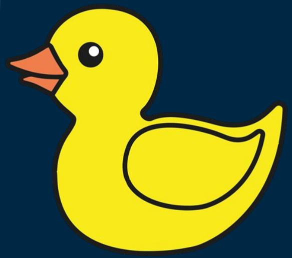
Train a classification model



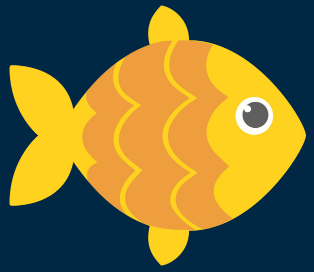
Train a classification model



Train a classification model



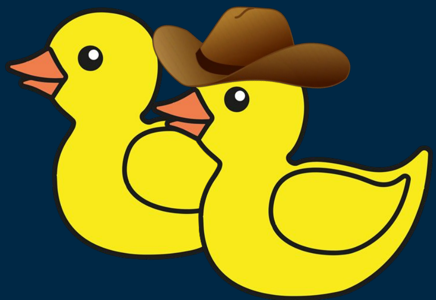
VS



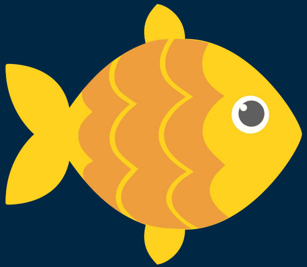
vs

Model 2

Train a classification model



VS



vs

Model 3

How can we improve AI efficiency, scalability and adaptability?

*(Hence making it **sustainable in the long term**)*

Continual Learning

Continual Learning

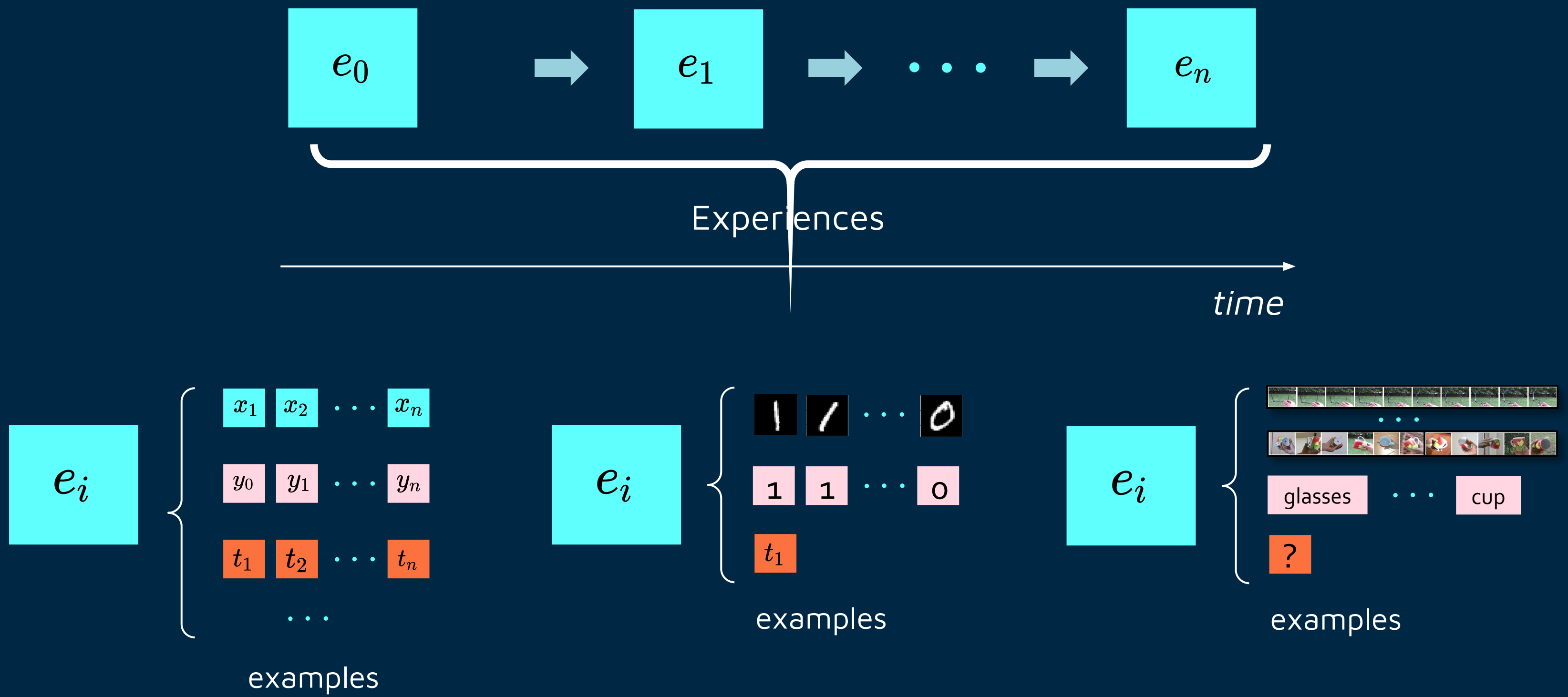
Continual learning is a set of approaches to train machine learning models incrementally, using data samples only once as they arrive.

Continual Learning Desiderata

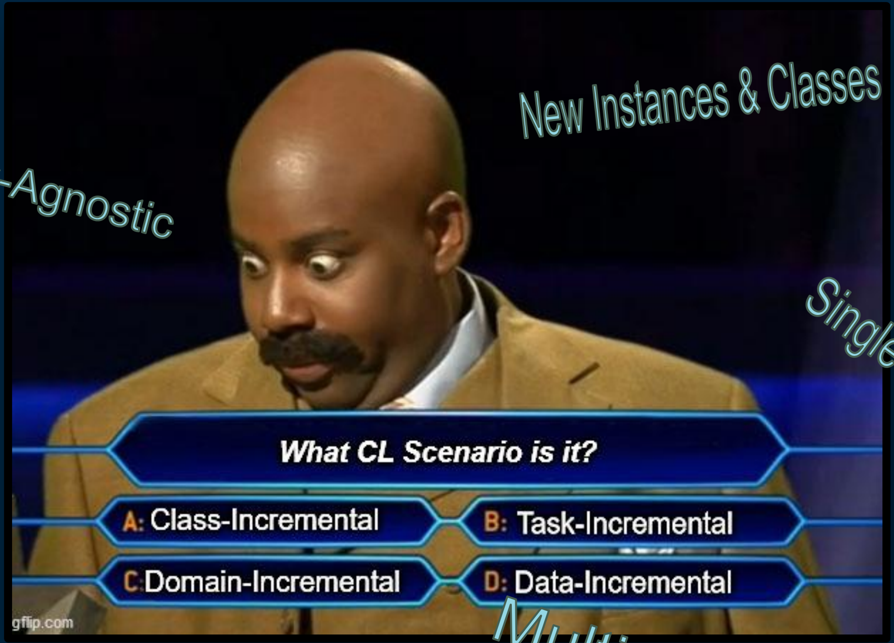
- Higher and **realistic time-scale** where data (and tasks) become available only during time.
- **No access** to previously encountered data.
- **Constant** computational and memory resources (efficiency)
- **Incremental development** of ever more complex knowledge and skills (scalability)
- **Efficiency** + **Scalability** = **Sustainability**

Sustainable Artificial Intelligence through Continual Learning. Cossu, A., Ziosi, M., Lomonaco, V., International Conference on AI for People (CAIP), 2021.

Continual Learning



Continual Learning Scenarios



Task-Agnostic

Task-Free

Single-Incremental-Task

Multi-Task

A Possible Categorization

Experience content type

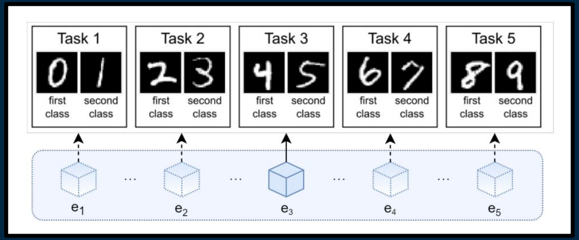
Task Labels

	New Instances (NI)	New Classes (NC)	New Instances and Classes (NIC)
Multi-Task	-	Task Incremental	-
Single-Incremental-Task	Domain-Incremental	Class-Incremental	Data-Incremental
Multiple-Incremental-Task	?	?	?

Our Dataset & Scenario

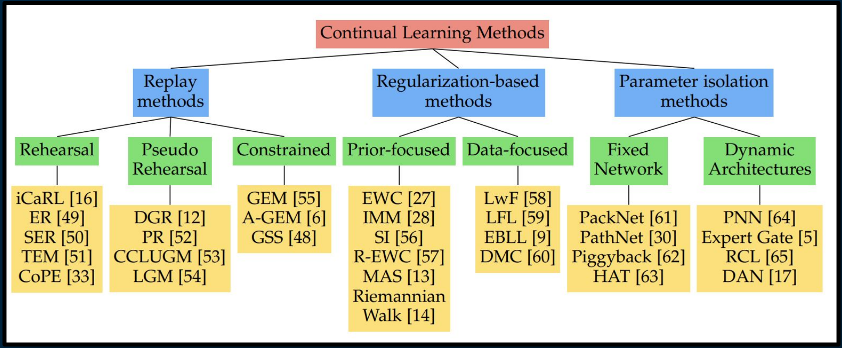
Mnist Dataset

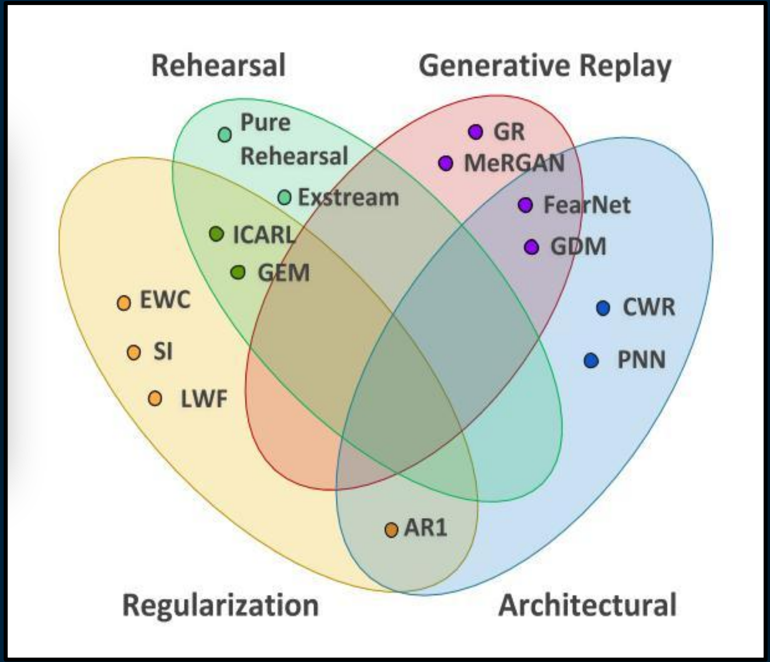
Split Mnist Benchmark



Continual Learning Strategy

Possible 4-way Categorization



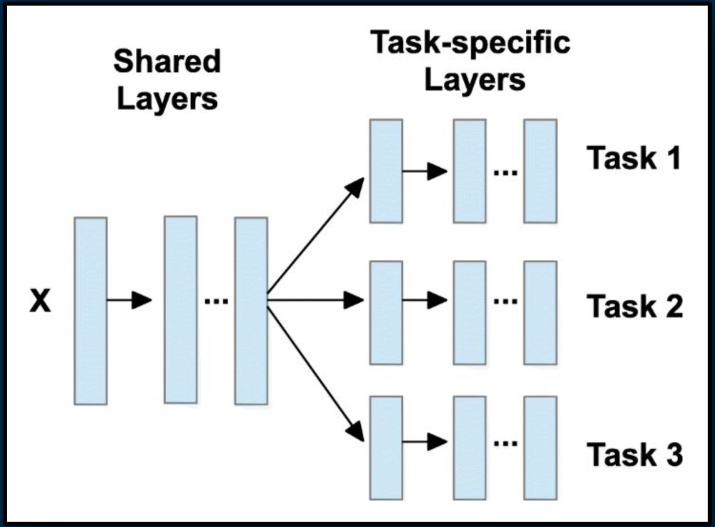


Continual Learning for Robotics: Definition, Framework, Learning Strategies, Opportunities and Challenges, Lesort et al. Information Fusion, 2020.
A continual learning survey: Defying forgetting in classification tasks. De Lange et al, TPAMI 2021.

Continual Learning Baselines

Common Baselines / Control Algorithms

- **Naive** / **Finetuning** (just continuing backprop)
- **Ensemble**: one model for each experience
- **Cumulative**: for every experience, accumulate all data and re-train from scratch.

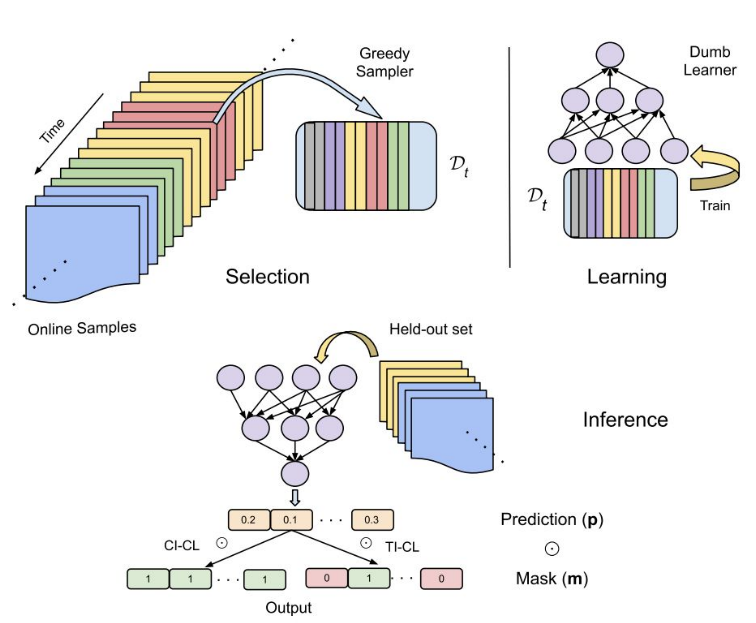


A brief review on multi-task learning. Thung et al, 2018.

GDUMB: Replay based approach

Greedy Sampler and Dumb Learner

- Interesting paper that sparked **strong discussions** in the CL community
- Note that there's **no knowledge transfer** in this strategy (quite dumb indeed!)
- Despite its simplicity, It was shown to work better than some existing and more complex strategies, **questioning the utility of some benchmarks/metrics** in our field



[GDumb: A Simple Approach that Questions Our Progress in Continual Learning](). Prabhu et al. ECCV, 2020.

Learning Without Forgetting (LWF)

Key Aspects

- Straightforward application of **Knowledge Distillation**
- Originally designed for **Task-Incremental settings** can be easily extended to others
- **Efficient single-head implementations** exist
- **Easy to implement** and commonly used

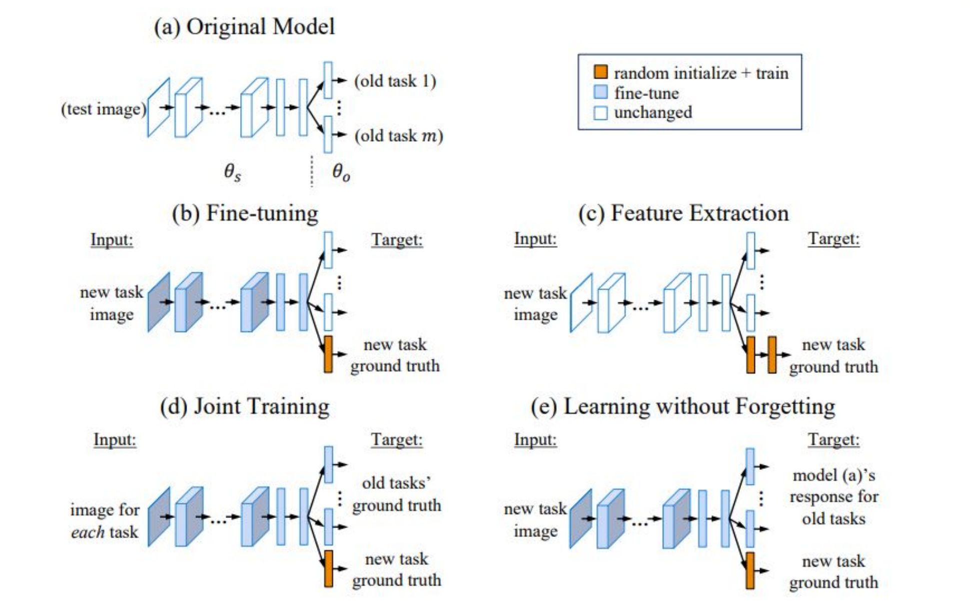


Fig. 2. Illustration for our method (e) and methods we compare to (b-d). Images and labels used in training are shown. Data for different tasks are used in alternation in joint training.

Learning without Forgetting, Li et al, TPAMI 2017
Distilling the knowledge in a neural network, Hinton et al, 2015.
Continuous Learning in Single-Incremental Tasks, Maltoni & Lomonaco, Neural Networks, 2019.

Copy Weights with Re-Init (CWR)

Algorithm 1 CWR* pseudocode: $\bar{\Theta}$ are the class-shared parameters of the representation layers; the notation $cw[j]$ / $tw[j]$ is used to denote the groups of consolidated / temporary weights corresponding to class j. Note that this version continues to work under NC, which is seen here a special case of NIC; in fact, since in NC the classes in the current batch were never encountered before, the step at line 7 loads 0 values for classes in B_i because cw_j were initialized to 0 and in the consolidation step (line 13) $wpast_j$ values are always 0.

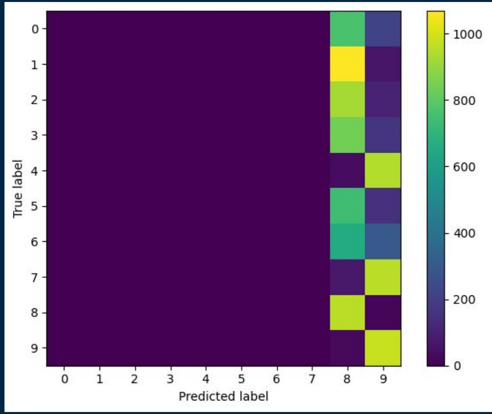
1: **procedure** CWR*
2: $cw = 0$
3: $past = 0$
4: init $\bar{\Theta}$ random or from pre-trained model (e.g. on ImageNet)
5: **for each** training batch B_i:
6: expand output layer with neurons for the new classes in B_i never seen before
7: $tw[j] = \begin{cases} cw[j], & \text{if class } j \text{ in } B_i \\ 0, & \text{otherwise} \end{cases}$
8: train the model with SGD on the s_i classes of B_i:
9: **if** $B_i = B_1$ learn both $\bar{\Theta}$ and tw
10: **else** learn tw while keeping $\bar{\Theta}$ fixed
11: **for each** class j in B_i:
12: $wpast_j = \sqrt{\frac{past_j}{cur_j}}$, where cur_j is the number of patterns of class j in B_i
13: $cw[j] = \frac{cw[j] \cdot wpast_j + (tw[j] - avg(tw))}{wpast_j + 1}$
14: $past_j = past_j + cur_j$
15: test the model by using $\bar{\Theta}$ and cw

Key Aspects

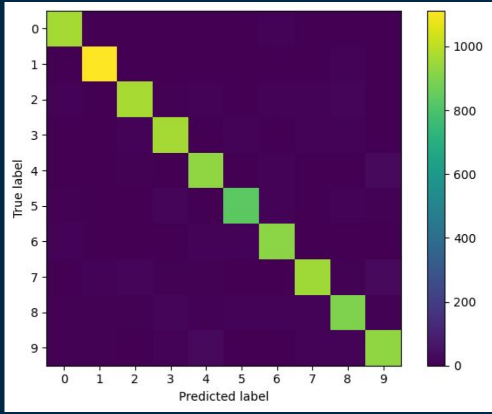
- Developed for the **fully connected linear classifier** (may be extended to multiple layers)
- **Dual memory system approach**: one for better plasticity, one for memory consolidation
- Very simple and efficient, yet effective solution **agnostic to the experience content (NI, NC, NIC) and specific scenario**

Rehearsal-Free Continual Learning over Small Non-I.I.D. Batches. Lomonaco et al, CLVision Workshop at CVPR 2020.

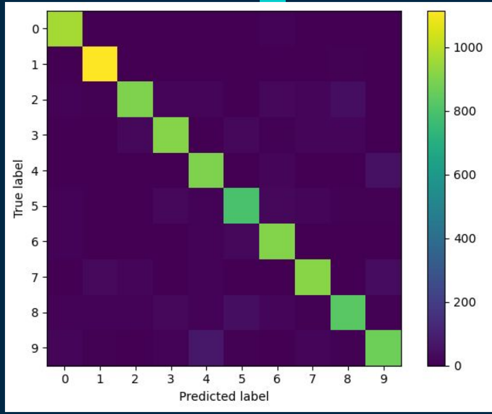
Results (Accuracy Matrix)



Naive

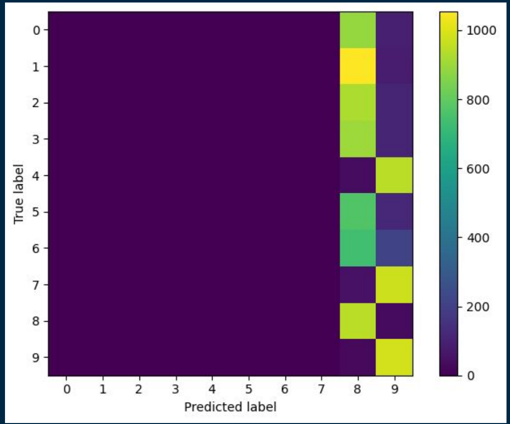


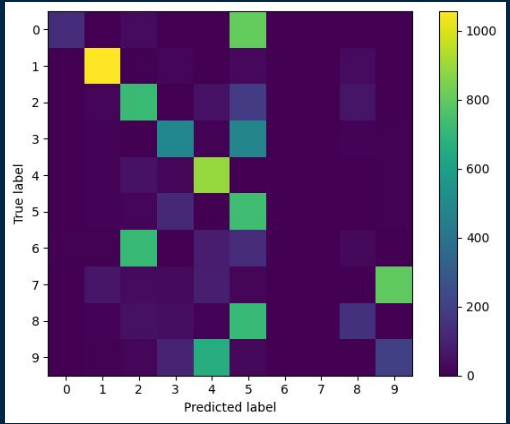
Cumulative



GDumb

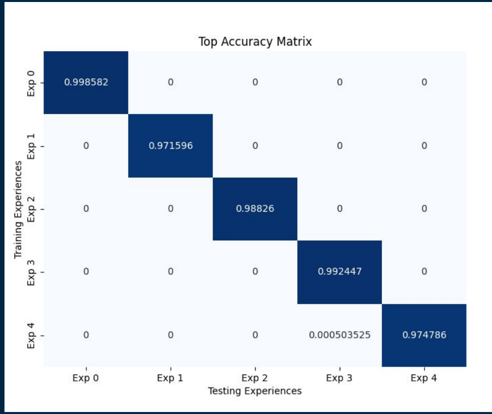
LWF



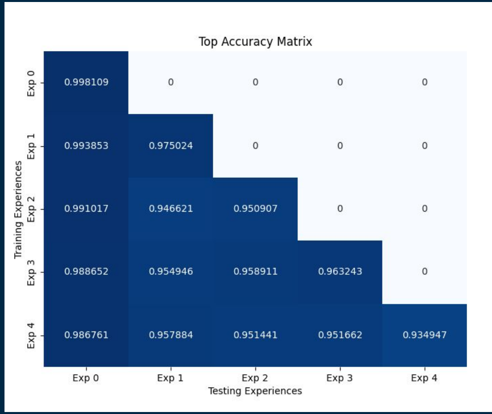


CWR

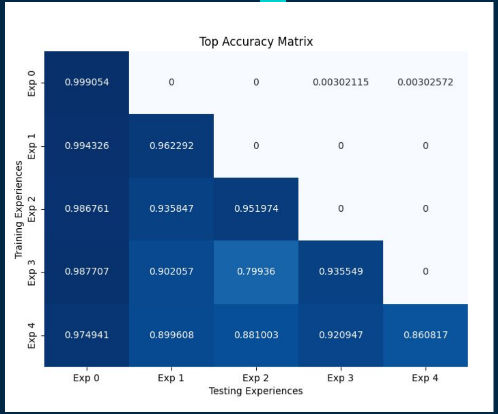
Results (Accuracy)



Naive

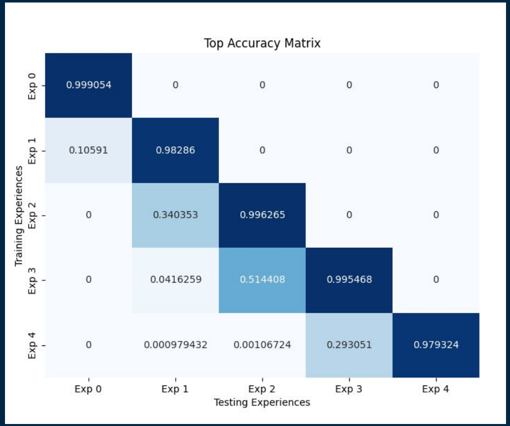


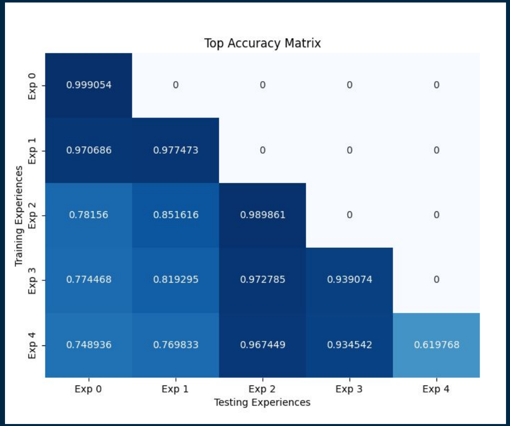
Cumulative



GDumb

LWF





CWR

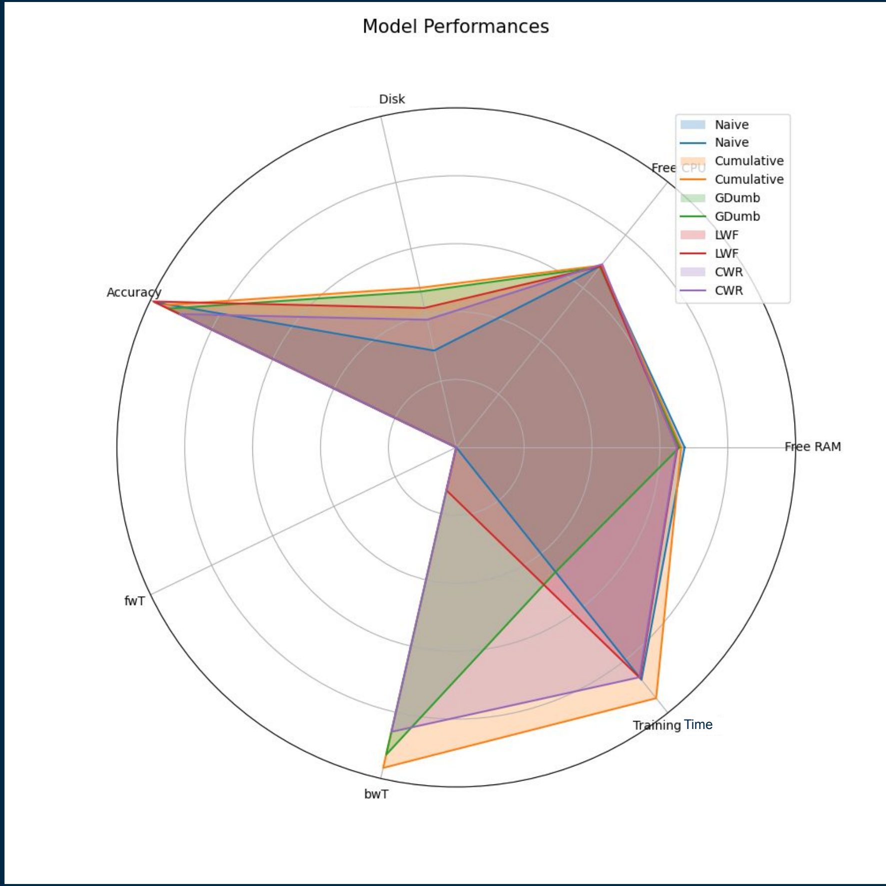
Accuracy metrics

$$\textbf{Backward Transfer} = \frac{\sum_{i>j}^{N} \mathcal{R}_{i,j}}{\frac{N(N-1)}{2}}$$

$$\textbf{Forward Transfer} = \frac{\sum_{i<j}^{N} \mathcal{R}_{i,j}}{\frac{N(N-1)}{2}}$$

$$\textbf{In-Domain Accuracy} = \frac{\sum_{i=1}^{N} \mathcal{R}_{i,i}}{N}$$

Results (Radar Graph)



Future Works

Key Aspects

- Experiment with hybrid models like AR1
- Benchmark on complex datasets like CLEAR (Continual LEArning on Real-World Imagery) dataset
- Explore implementation in robot vision related tasks. I.e. ego-centric video datasets

Q & A